TURBO-CHARGER SURGE DETECTION

BACKGROUND OF THE INVENTION

Cross-reference to Related Applications:

[0001] This application claims priority to Provisional Application Serial No. 60/518,648, filed November 12, 2003, the disclosure of which is incorporated by reference.

Field of the Invention:

[0002] This invention relates to the field of turbo-charger surge detection.

Description of the Related Art:

[0003] An internal combustion engine may include a super-charger or a turbo-charger for compressing intake air prior to delivery to the combustion chambers. A super-charger is typically belt or gear-driven, while a turbo-charger has a turbine which is driven by the engine's exhaust gases. The belt or gears, in the case of a supercharger, or the turbine, in the case of a turbocharger, drives a compressor which compresses the intake air. The compressor, which may be a centrifugal or rotary pump, receives air to be compressed on an inlet side and supplies the air to the combustion chambers from an outlet side. The difference in pressure of the outlet side relative to the inlet side is termed the pressure ratio, and represents the amount of boost the compressor is supplying to the intake air.

[0004] In Fig. 1 is shown a compressor map 100 for a turbocharger compressor. Pressure ratios 102 are plotted on the vertical axis and rates of flow 104 are plotted on the horizontal axis. As may be seen in Fig. 1, an operating region 106 of the compressor is bounded on the left side of the compressor map by a surge line 108.

[0005] Surge occurs when rate of flow 104 through the compressor is too small to support the prevailing pressure ratio 102. Surge line 108 represents this condition for various rates of flow 104. When rate of flow 104 is too small to support the prevailing pressure ratio 102, the air flow will cavitate, separating from the suction side of the blades or vanes of the compressor wheel and reversing air flow through the compressor until pressure ratio 102 is reduced. If the surge conditions continue to prevail, pressure ratio 102 will build up again and the cycle will be repeated. This cycle of rising and falling pressure ratios 102 may continue at a substantially fixed frequency. Surge makes a popping noise and stresses the piping between the turbo-charger and the inlet to the engine. The popping noise is called surging or barking. Customers using a truck that has this surging or barking dislike it and are afraid that it is causing damage to their truck and the engine.

[0006] The interaction between the turbo-charger, such as a variable geometry turbo-charger, and an EGR system may exacerbate a back flow of gas through the turbo-charger compressor. An EGR system may provide exhaust gas downstream of the compressor to avoid soaking the compressor in corrosive exhaust gases. The recirculated exhaust gas entering the air flow downstream of the compressor may add to the pressure at the outlet of the compressor, raising pressure ratio 102 artificially and promoting surge.

SUMMARY OF THE INVENTION

[0007] A primary object of the invention is to overcome the deficiencies of the related art described above by providing a turbo-charger surge detection method

and system. The present invention achieves these objects and others by providing a turbo-charger surge detection method and system.

[0008] In several aspects, the invention may provide a turbo-charger surge detection method and system. In particular, in a first aspect, a method of turbo-charger surge detection may include the steps of measuring a rate of air flow through a turbo-charger compressor, measuring a temperature of the air flow, calculating a standard mass flow rate of the air flow at the rate and the temperature, measuring a pressure ratio across the turbo-charger compressor, calculating a surge mass flow rate at a surge line of the compressor at the pressure ratio, comparing the standard mass flow rate to the surge mass flow rate, and reducing an EGR flow if the standard mass flow rate is lower than the surge mass flow rate.

[0009] In a second aspect, a method of turbo-charger surge detection may include the steps of measuring a rate of air flow through a turbo-charger compressor, measuring a temperature of the air flow, calculating a standard mass flow rate of the air flow at the rate and the temperature, measuring a pressure ratio across the turbo-charger compressor, calculating a surge mass flow rate at a surge line of the compressor at the pressure ratio, comparing the standard mass flow rate to the surge mass flow rate, and reducing the pressure ratio by opening a vane of the compressor if the standard mass flow rate is lower than the surge mass flow rate.

[0010] In a third aspect, a system for turbo-charger surge detection may include means for measuring a rate of air flow through a turbo-charger compressor, means for measuring a temperature of the air flow, means for calculating a standard mass flow rate of the air flow, means for measuring a pressure ratio across the turbo-charger compressor, means for calculating a surge mass flow rate at a surge line of

the compressor, means for comparing the standard mass flow rate to the surge mass flow rate, and means for reducing an EGR flow if the standard mass flow rate is lower than the surge mass flow rate.

[0011] In a fourth aspect, a system for turbo-charger surge detection may include means for measuring a rate of an air flow through a turbo-charger compressor, means for measuring a temperature of the air flow, means for calculating a standard mass flow rate of the air flow, means for measuring a pressure ratio across the turbo-charger compressor, means for calculating a surge mass flow rate at a surge line of the compressor, means for comparing the standard mass flow rate to the surge mass flow rate, and means for reducing the pressure ratio by opening a vane of the compressor.

[0012] The above and other features and advantages of the present invention, as well as the structure and operation of various embodiments of the present invention, are described in detail below with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

[0013] The accompanying drawings, which are incorporated herein and form part of the specification, illustrate various embodiments of the present invention and, together with the description, further serve to explain the principles of the invention and to enable a person skilled in the pertinent art to make and use the invention. In the drawings, like reference numbers indicate identical or functionally similar elements. A more complete appreciation of the invention and many of the attendant advantages thereof will be readily obtained as the same becomes better understood

by reference to the following detailed description when considered in connection with the accompanying drawings, wherein:

[0014] Fig. 1 is compressor map;

Fig. 2 is a schematic diagram of a turbo-charged internal combustion engine for use with an embodiment of the invention;

Fig. 3 is a schematic diagram of a control algorithm according to a first embodiment of the invention; and

Fig. 4 is compressor map for use with an embodiment of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0015] In Fig. 2 is shown a schematic diagram of a turbo-charged internal combustion engine 222 for use with an embodiment of the invention. Turbo-charged internal combustion engine 222 may include a plurality of cylinders 224, each having a combustion chamber 226 fed by a runner 228 of an intake manifold 230. A compressor 204 may provide pressurized intake air 202 to intake manifold 230. Compressor 204 may have an inlet 240 receiving low pressure air 202, which may be at ambient pressure, and an outlet 242 plumbed to intake manifold 230. Also plumbed to intake manifold 230 may be an outlet 232 of an EGR valve 234. An inlet 236 of EGR valve 234 may scavenge exhaust gases from an exhaust manifold 238 also connected to combustion chambers 226.

[0016] It would be desirable if a vane position of compressor 204 could be adjusted when compressor 204 was at the point of surging to lower the pressure ratio across compressor 204 and avert surge. It would further be desirable if EGR flow into intake manifold 230 downstream from the compressor could be reduced

when compressor 204 was at the point of surging to lower the pressure ratio across compressor 204 and avert surge.

[0017] In Fig. 3 is shown a schematic diagram 244 of a control algorithm for turbo-charged internal combustion engine 222. Input parameters 246 shown on the left side of the diagram 244 are evaluated to determine where compressor 204 is operating relative to surge line 108. An actual mass air flow 202 through compressor 204 and a temperature 206 are used to calculate a standard mass flow rate 208 at the reference conditions for compressor 204 (mdotcorr). A pressure ratio 210 of compressor 204 is determined from the input sensors (CPR) and used to lookup a mass flow rate 212 at compressor 204's surge line 108. The look up curve has some surge margin 216 added, as shown in Fig. 4.

[0018] Standard mass flow rate 208 at compressor 204 reference condition is compared to mass flow rate 212 at surge line 108 (plus surge margin 216) to determine if compressor 204 is close to surge. After this comparison, a controller 250 may reduce an EGR flow 214 proportionally to force more mass air flow 202 through compressor 204, thus eliminating surging of compressor 204 when no surge margin 216 is available. Otherwise the desired EGR flow request is passed directly without any reduction. Modifying EGR valve 234 position to reduce an EGR flow 214 may improve air flow 202 through compressor 204, thus improving surge margin 216.

[0019] Pressure ratio 210 of compressor 204 can also be lowered by opening the turbo-charger compressor 204 vane 218 position to further improve surge margin 216. Any combination of events can be used to improve surge margin 216 of compressor 204. Modify EGR valve 234 position only, modify EGR valve 234

position first followed by opening turbo-charger vane 218 position, opening turbo-charger vane 218 position first followed by modifying EGR valve 234 position, or only opening turbo-charger vane 218 position.

[0020] Two additional cases for turbo-charger interaction with the EGR system were found under rapid deceleration of the engine either in braking mode or without braking.

Case 1: EGR valve 234 must be rapidly closed to prevent back flow of gas through compressor 204. The natural action of the EGR system with a quick removal of fuel rate is to increase the EGR flow rate 214 to maintain the desired air to fuel ratio. The turbo-charger is still delivering a high flow rate of air 202 without the fuel to go with it. The EGR system will try to compensate with a higher amount of EGR thus reducing the flow of air 202 through compressor 204. This leads to a back flow of gas through compressor 204 because there is not enough power left to spin the turbo-charger at the given pressure ratio 210. The turbo-charger vane 218 position can also be reduced in combination with EGR valve 234 closure to remove the demand for compressor 204 as a function of a quick removal of fuel rate.

[0021] Case 2: The turbo-charger vane 218 position may also be modified quickly in the case of rapid deceleration of the engine with the brake applied. Energy must be removed from the system in a quick controlled manner to prevent the barking (popping noise) of the turbo-charger. Braking mode requires less flow rate of air 202 compared to the powered operation flow rate of air 202. A back flow of gas through compressor 204 occurs with the rapid change in air usage.

[0022] In particular, in a first embodiment, a method of turbo-charger surge detection 200 may include the steps of measuring a rate 220 of air flow 202 through

a turbo-charger compressor 204, measuring a temperature 206 of air flow 202, calculating a standard mass flow rate 208 of air flow 202 at rate 220 and temperature 206, measuring a pressure ratio 210 across turbo-charger compressor 204, calculating a surge mass flow rate 212 at a surge line 214 of compressor 204 at pressure ratio 210, comparing standard mass flow rate 208 to surge mass flow rate 212, and reducing an EGR flow 214 if standard mass flow rate 208 is lower than surge mass flow rate 212.

In one embodiment, the method of turbo-charger surge detection may also include the step of adding a surge margin 216 to surge mass flow rate 212.

[0023] In one embodiment, method of turbo-charger surge detection 200 may also include the step of reducing pressure ratio 210 by opening a vane 218 of compressor 204.

[0024] In a second embodiment, a method of turbo-charger surge detection 300 may include the steps of measuring rate 220 of air flow 202 through turbo-charger compressor 204, measuring temperature 206 of air flow 202, calculating a standard mass flow rate 208 of air flow 202 at rate 220 and temperature 206, measuring a pressure ratio 210 across turbo-charger compressor 204, calculating a surge mass flow rate 212 at a surge line 214 of compressor 204 at pressure ratio 210, comparing standard mass flow rate 208 to surge mass flow rate 212, and reducing pressure ratio 210 by opening a vane 218 of compressor 204 if standard mass flow rate 208 is lower than surge mass flow rate 212.

[0025] In one embodiment, method of turbo-charger surge detection 300 may also include the step of adding a surge margin 216 to surge mass flow rate 212.

[0026] Example I: An example of a turbo-charger surge detection program for use with an embodiment of the invention follows.

EXAMPLE I

1 EGR_Step_7_Components

Library	:	Project Library	EGR_Step_7_Components
Entry	:	Project	kicker

1.1 kicker

Path:

EGR_Step_7_Components/kicker

Notes:

1.1.1 Codegeneration Options

Code Generator	:	Physical Experiment
Expander	:	ANSI-C for Experimental Targets
protected division	:	true
data logging	:	false
protected vector indices	:	true
optimize direct access methods (one level)	:	false
optimize direct access methods (multiple levels)	:	false
use PMI when generating c-code components	:	true
generate access methods for dT (alternative: use OS dT directly)	:	true
Internal Make - generate Map file	:	true
Internal Make - updating dependent parameters	:	true
Internal Make - use long file names	:	false
Internal Make - generate one global header file (temp.h)	:	true
Internal Make - keep generic sources for external make	:	false
max number of loop iterations	:	1000
warning level (between 0 [no] and 2 [all])	:	1

1.1.2 Target Options

Target	:	ES1112
Tool	:	DIABDATAV41x

1.1.3 Operating System

Cooperative Level	:	19
Preemptive Level	:	7
Operating Modes	:	inactive[0], active[1]

1.1.4 Settings for Task

Name	Trigger Mode	Priority	Priority Group	Delay[s]	Period[s]
Init	Init	—	—	—	0.005
Exit	Init	—	—	—	0.005
ETK_A	Software	17	preemptive	—	0.005
ETK_B	Software	16	preemptive	—	0.005
Config	Timer	2	cooperative	0.0	0.5

Name	Trigger Event	Max. Act.	Operating Modes	Monitoring
Init	—	—	active[1] [START]	false
Exit	—	—	inactive[0]	false
ETK_A	—	1	active[1] [START]	false
ETK_B	—	1	active[1] [START]	false
Config	—	1	active[1] [START]	false

1.1.5 Task Schedule

Init		Exit	
Process	Class	Process	Class
Es1000usap_InitCode_Init_HWCF	HWC	Etkc_ExitCode_Exit_HWCL	HWC
Etkc_InitCode_Init_HWCF	HWC	-	-
Init	surge_protect	-	-

ETK_A		ETK_B	
Process	Class	Process	Class
Etk_bypassbypass_n0_receive_ETK_A_HWCF	HWC	Etk_bypassbypass_t10_receive_ETK_B_HWCF	HWC
Etk_bypassbypass_n0_send_ETK_A_HWCL	HWC	outtimeslot	EGOO_EGR_OnOff
-	-	process	surge_protect
-	-	Etk_bypassbypass_t10_send_ETK_B_HWCL	HWC

1.1.6 Global Variables

Name	Value	Unit	Modeltype	Kind	Dimension
derp_r_w	0.0	—	mesg[cont]	var	scalar
dT	0.0	—	dT	var	scalar
eahs_r_kcgin_w	0.0	—	mesg[cont]	var	scalar
eahs_r_kgcln_w	0.0	—	mesg[cont]	var	scalar
eahs_t_ambtemp_w	0.0	—	mesg[cont]	var	scalar
eams_dm_prsrlf_w	0.0	—	mesg[cont]	var	scalar
eapp_dpv_w	0.0	—	mesg[cont]	var	scalar
eapp_pv_w	0.0	—	mesg[cont]	var	scalar
eaps_p_w	0.0	—	mesg[cont]	var	scalar
eats_t_w	0.0	—	mesg[cont]	var	scalar
ebps_p_w	0.0	—	mesg[cont]	var	scalar
ebts_t_w	0.0	—	mesg[cont]	var	scalar

ecdt_t_w	0.0	---	mesg[cont]	var	scalar
ects_t_w	0.0	---	mesg[cont]	var	scalar
eess_n_avg_w	0.0	---	mesg[cont]	var	scalar
egdm_dm_aftsteplim_w	0.0	---	mesg[cont]	var	scalar
egdm_dm_des_w	0.0	---	mesg[cont]	var	scalar
egdm_dm_desinterm_w	0.0	---	mesg[cont]	var	scalar
egfm_dm_atk_w	0.0	---	mesg[cont]	var	scalar
egfm_dm_egrout_w	0.0	---	mesg[cont]	var	scalar
egfm_dm_gasatj_w	0.0	---	mesg[cont]	var	scalar
egfm_p_map_w	0.0	---	mesg[cont]	var	scalar
egrt_t_w	0.0	---	mesg[cont]	var	scalar
egvc_pos_govout_w	0.0	---	mesg[cont]	var	scalar
fqsc_q_w	0.0	---	mesg[cont]	var	scalar
fqsc_r_netload_w	0.0	---	mesg[cont]	var	scalar
fqsc_s_s	0.0	---	mesg[cont]	var	scalar

Name	Memory	Comment
derp_r_w	v	---
dT	v	---
eahs_r_kcgin_w	v	---
eahs_r_kgcin_w	v	---
eahs_t_ambtemp_w	v	---
eams_dm_prsrlf_w	v	---
eapp_dpv_w	v	---
eapp_pv_w	v	---
eaps_p_w	v	---
eats_t_w	v	---
ebps_p_w	v	---
ebts_t_w	v	---
ecdt_t_w	v	---
ects_t_w	v	---
eess_n_avg_w	v	---
egdm_dm_aftsteplim_w	v	---
egdm_dm_des_w	v	---
egdm_dm_desinterm_w	v	---
egfm_dm_atk_w	v	---
egfm_dm_egrout_w	v	---
egfm_dm_gasatj_w	v	---
egfm_p_map_w	v	---
egrt_t_w	v	---
egvc_pos_govout_w	v	---
fqsc_q_w	v	---
fqsc_r_netload_w	v	---
fqsc_s_s	v	---

1.1.7 Implementation of Global Variables

Name	Model(M)				Formula	Implementation(I)		
	Type	Min.	Max.	Q.	M=a*I+b	Type	Min.	Max.
derp_r_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
dT	cont	0.0	2147.483	0	dT	uint32	0	16777211
eahs_r_kcgin_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
eahs_r_kgcin_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
eahs_t_a	cont	-oo	+oo	0	ident	real64	-oo	+oo

mbtemp_w								
eams_dm_prsrlf_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
eapp_dpv_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
eapp_pv_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
eaps_p_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
eats_t_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
ebps_p_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
ebts_t_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
ecdt_t_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
ects_t_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
eess_n_avg_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
egdm_dm_aftsteplim_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
egdm_dm_des_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
egdm_dm_desinterm_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
egfm_dm_atk_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
egfm_dm_egrout_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
egfm_dm_gasatj_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
egfm_p_map_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
egrt_t_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
egvc_pos_govout_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
fqsc_q_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
fqsc_r_netload_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
fqsc_s_s	cont	-oo	+oo	0	ident	real64	-oo	+oo

1.1.8 Data

Name: Data

Notes:

—

Elements:

Module	Data	Module	Data
EGOO_EGR_OnOff	Data	HWC	Data
surge_protect	Data		

1.1.9 Implementations

Name: Impl

Notes:
—

Elements:

Module	Implementation	Module	Implementation
EGOO_EGR_OnOff	Impl	HWC	Impl
surge_protect	Impl		

1.1.10 Project Formulas

Name	Type	Formula (M=a*I+b)
ident	Identity	f(phys) = phys

2 turbokicker

2.1 Decel_Catch

Library : Component Library turbokicker
Entry : Component Decel_Catch

Path:

turbokicker/Decel_Catch

Notes:
—

2.1.1 Layout

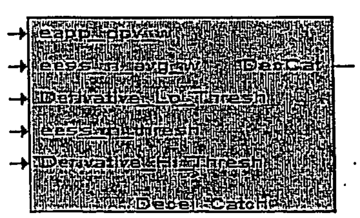

Figure 1: Layout of class Decel_Catch

2.1.2 Public Methods

DecCat

Name	Type	Unit	Comment	Kind
DecCat/eapp_dpv_w	cont	%/sec	Pedal derivative rate	arg
DecCat/eess_n_avg_w	cont	RPM	Engine Speed	arg
DecCat/Derivative_Lo_Thresh	cont	%/sec	Activation threshold	arg
DecCat/eess_n_thresh	cont	RPM	engine speed activation threshold	arg
DecCat/Derivative_Hi_Thresh	cont	%/sec	Deactivation threshold	arg
DecCat/return	log	---	EGOO output bit	ret
EGOO_Bit_Status/DecCat	log	---	Status of EGOO bit	local

2.1.3 ESDL Description

```
DecCat
[eapp_dpv_w::cont;eess_n_avg_w::cont;Derivative_Lo_Thresh::cont;eess_n_thresh::c
ont;Derivative_Hi_Thresh::cont] return::log
if ((eess_n_avg_w < eess_n_thresh) && (eapp_dpv_w < Derivative_Lo_Thresh))
        EGOO_Bit_Status = false;

if ((eess_n_avg_w < eess_n_thresh) && (eapp_dpv_w > Derivative_Hi_Thresh) &&
(!EGOO_Bit_Status))
        EGOO_Bit_Status = true;

if (eess_n_avg_w > eess_n_thresh)
        EGOO_Bit_Status = true;

return EGOO_Bit_Status;
```

2.1.4 Implementations

Name: Impl

Notes:

Elements:

Name	Model(M)				Formula	Implementation(I)		
	Type	Min.	Max.	Q.	M=a*I+b	Type	Min.	Max.
DecCat/Derivative_Hi_Thresh	cont	-oo	+oo	0	ident	real64	-oo	+oo
DecCat/Derivative_Lo_Thresh	cont	-oo	+oo	0	ident	real64	-oo	+oo
DecCat/eapp_dpv_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
DecCat/eess_n_av	cont	-oo	+oo	0	ident	real64	-oo	+oo

g_w								
DecCat/eess_n_thresh	cont	-oo	+oo	0	ident	real64	-oo	+oo

Name	Implementation_type	Name	Implementation_type
DecCat/return	int8		

2.2 press_ratio

Library : Component Library turbokicker
Entry : Component press_ratio

Path:

turbokicker/press_ratio

Notes:

2.2.1 Layout

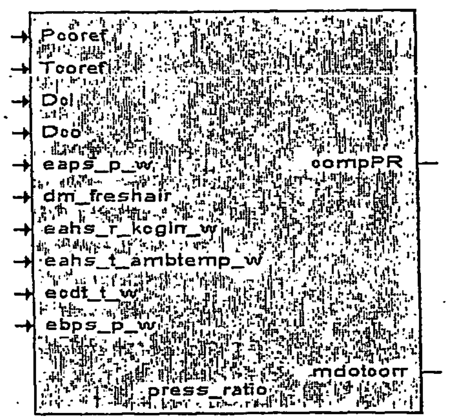

Figure 2: Layout of class press_ratio

2.2.2 Elements

Name	Value	Unit	Modeltype	Kind	Dimension
MathFcn	Data	---	MathFcn	var	compl..
mdotTCC	0.0	lbs/min	cont	var	scalar

Name	Memory	Comment
MathFcn	v	---
mdotTCC	v	Corrected Compressor Flow

2.2.3 Public Methods

compPR

Name	Type	Unit	Comment	Kind
compPR/eaps_p_w	cont	psia	barometer	arg
compPR/dm_freshair	cont	lbs/min	fresh air flow	arg
compPR/eahs_r_kcgin_w	cont	—	humidity correction	arg
compPR/eahs_t_ambtemp_w	cont	deg F	inlet temp to compressor	arg
compPR/Dci	cont	inches	compressor inlet DIA	arg
compPR/ecdt_t_w	cont	deg F	compfressor discharge temp	arg
compPR/ebps_p_w	cont	psi	Boost pressure	arg
compPR/Dco	cont	inches	compressor outlet DIA	arg
compPR/Pcoref	cont	psi	Compressor map ref pressure	arg
compPR/Tcoref	cont	deg R	Compressor map ref temp	arg
compPR/return	cont	press ratio	compressor pressure ratio	ret
Pdi/compPR	cont	psi	dyn pressure comp inlet	local
Pdcd/compPR	cont	psi	dyn pressure comp outlet	local

mdotcorr

Name	Type	Unit	Comment	Kind
mdotcorr/return	cont	lb/min	correct mass flow	ret

2.2.4 ESDL Description

```
compPR
[eaps_p_w::cont;dm_freshair::cont;eahs_r_kcgin_w::cont;eahs_t_ambtemp_w::cont;Dc
i::cont;ecdt_t_w::cont;ebps_p_w::cont;Dco::cont;Pcoref::cont;Tcoref::cont]
return::cont
/* Compressor map corrections ussually use inlet restrictions
 * as part of the results.  Since no inlet restriction value
 * is available, inlet restriction value is not used as part
 * of the value for pressure or mass flow corrections.*/

/* calculation of dynamic pressure at compressor inlet */

Pdi = (.00007581028 * eahs_r_kcgin_w * dm_freshair * dm_freshair *
(eahs_t_ambtemp_w + 459.67)) / ((eaps_p_w * 2.0416) * Dci*Dci*Dci*Dci);

/* calculation of dynamic pressure at compressor outlet */

Pdcd = (.00007581028 * eahs_r_kcgin_w * dm_freshair * dm_freshair * (ecdt_t_w +
459.67)) / (((eaps_p_w + ebps_p_w) * 2.0416) * Dco*Dco*Dco*Dco);
```

```
/* calculation of corrected mass flow through compressor*/

mdotTCC = dm_freshair * (Pcoref/(Pdi + eaps_p_w)) *
MathFcn.sqrt((eahs_t_ambtemp_w + 459.67)/Tcoref);

/* calculate compressor pressure ratio*/

return (ebps_p_w + Pdcd + eaps_p_w) / (Pdi + eaps_p_w);


mdotcorr [] return::cont
return (mdotTGC);
```

2.2.5 Implementations

Name: Impl

Notes:

Elements:

Name	Model(M)				Formula	Implementation(I)		
	Type	Min.	Max.	Q.	M=a*I+b	Type	Min.	Max.
compPR/Dci	cont	-oo	+oo	0	ident	real64	-oo	+oo
compPR/Dco	cont	-oo	+oo	0	ident	real64	-oo	+oo
compPR/dm_freshair	cont	-oo	+oo	0	ident	real64	-oo	+oo
compPR/eahs_r_kcgin_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
compPR/eahs_t_ambtemp_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
compPR/eaps_p_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
compPR/ebps_p_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
compPR/ecdt_t_w	cont	-oo	+oo	0	ident	real64	-oo	+oo
compPR/Pcoref	cont	-oo	+oo	0	ident	real64	-oo	+oo
compPR/return	cont	-oo	+oo	0	ident	real64	-oo	+oo
compPR/Tcoref	cont	-oo	+oo	0	ident	real64	-oo	+oo
mdotcorr/return	cont	-oo	+oo	0	ident	real64	-oo	+oo
mdotTCC	cont	-oo	+oo	0	ident	real64	-oo	+oo

2.3 surge_protect

Library : Component Library turbokicker
Entry : Module surge_protect

Path:

turbokicker/surge_protect

Notes:

2.3.1 Layout

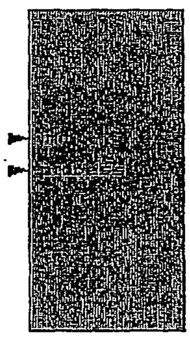

Figure 3: Layout of Module surge_protect

2.3.2 Elements

Name	Value	Unit	Modeltype	Kind	Dimension
cont	1048576.0	---	cont	par	scalar
CPR	0.0	---	cont	var	scalar
Dci	3.375	---	cont	par	scalar
Dco	2.875	---	cont	par	scalar
DecCat_output	false	---	log	var	scalar
Decel_Catch	Data	---	Decel_Catch	var	compl.
derv	0.0	---	cont	par	scalar
Derv_Hi_thresh	-90.0	---	cont	par	scalar
Derv_Lo_thresh	-100.0	---	cont	par	scalar
dm_at_surge	0.0	lb/min	cont	var	scalar
dm_freshair	0.0	---	cont	var	scalar
dm_surge_margin	0.0	---	cont	var	scalar
eess_n_thresh	1250.0	---	cont	par	scalar
egdm_s_enbarkprevent	true	---	log	par	scalar
EGOO_smoke_Active	false	---	log	var	scalar
enable_smoke_active	true	---	log	par	scalar
Hysteresis	HysteresisMDelta2	---	Hysteresis_MSP_DeltaHalf	var	compl.
initvalue	0.01	---	cont	par	scalar

intg	0.01	---	cont	par	scalar
intgmax	5.0	---	cont	par	scalar
intgmin	-5.0	---	cont	par	scalar
Limiter	Limiter	---	Limiter	var	compl.
maxlimit	0.0	---	cont	par	scalar
mdotcorr	0.0	lbs/min	cont	var	scalar
minlimit	-15.0	---	cont	par	scalar
Pcoref	14.5	psi	cont	par	scalar
PIDLimited	PID	---	PIDLimited	var	compl.
press_ratio	Data	---	press_ratio	var	compl.
remote_vpsp	0.0	---	cont	par	scalar
RMTC_DM_EGDM_CW	0.0	---	cont	par	scalar
rmtc_s_egdm	false	---	log	par	scalar
smoke_limit	252.0	---	cont	par	scalar
surge_line	s.u.	---	1D[cont->cont]	par	1-D table
Surge_Prot_eess_n_Lim	780.0	---	cont	par	scalar
surge_prot_K	s.u.	---	1D[cont->cont]	par	1-D table
Surge_Prot_Lim_Out	false	---	log	var	scalar
Surge_prot_n_delta2	20.0	---	cont	par	scalar
Tcoref	544.68	deg R	cont	par	scalar

Name	Memory	Comment
cont	nv	---
CPR	v	---
Dci	nv	---
Dco	nv	---
DecCat_output	v	---
Decel_Catch	v	---
derv	nv	---
Derv_Hi_thresh	nv	---
Derv_Lo_thresh	nv	---
dm_at_surge	v	---
dm_freshair	v	---
dm_surge_margin	v	---
eess_n_thresh	nv	---
egdm_s_enbarkprevent	nv	---
EGOO_smoke_Active	v	---
enable_smoke_active	nv	EGOO smoke active bit enable
Hysteresis	v	---
initvalue	nv	---
intg	nv	---
intgmax	nv	---
intgmin	nv	---
Limiter	v	---
maxlimit	nv	---
mdotcorr	v	mass flow at corrected compressor map conditions
minlimit	nv	---
Pcoref	nv	ref pressure for compressor map
PIDLimited	v	---
press_ratio	v	---
remote_vpsp	nv	---
RMTC_DM_EGDM_CW	nv	---
rmtc_s_egdm	nv	---
smoke_limit	nv	---
surge_line	nv	---
Surge_Prot_eess_n_Lim	nv	---
surge_prot_K	nv	---
Surge_Prot_Lim_Out	v	---
Surge_prot_n_delta2	nv	---

Tcoref	nv	Ref Temperature for Compressor map

2.3.3 Values for Elements

surge_prot_K

Type : OneDCharTable

x	650.0	900.0	1100.0	1200.0	1300.0	1400.0
w	0.25	0.5	1.5	2.0	2.0	1.5

x	1600.0	1800.0	1900.0
w	1.0	0.5	0.5

Interpolation Method : Linear
Extrapolation Method : Constant
Name of x Distribution : ---

surge_line

Type : OneDCharTable

x	0.0	1.555	1.962	2.431	2.948	3.483
w	0.0	12.541725	19.764045	28.4445	37.680825	44.08362

x	3.991
w	47.875275

Interpolation Method : Linear
Extrapolation Method : Constant
Name of x Distribution : ---

2.3.4 Imported Elements

Name	Value	Unit	Modeltype	Kind	Dimension
derp_r_w	---	---	mesg[cont]/R	var	scalar
eahs_r_kcgin_w	---	---	mesg[cont]/R	var	scalar
eahs_t_ambtemp_w	---	---	mesg[cont]/R	var	scalar
eams_dm_prsrlf_w	---	---	mesg[cont]/R	var	scalar
eapp_dpv_w	---	---	mesg[cont]/R	var	scalar
eapp_pv_w	---	---	mesg[cont]/R	var	scalar
eaps_p_w	---	---	mesg[cont]/R	var	scalar
ebps_p_w	---	---	mesg[cont]/R	var	scalar
ecdt_t_w	---	---	mesg[cont]/R	var	scalar
eess_n_avg_w	---	---	mesg[cont]/R	var	scalar
egdm_dm_aftsteplim_w	---	---	mesg[cont]/R	var	scalar
egdm_dm_des_w	---	---	mesg[cont]/R	var	scalar
egfm_dm_atk_w	---	---	mesg[cont]/R	var	scalar
egfm_dm_egrout_w	---	---	mesg[cont]/R	var	scalar
egoo_s_allowed_b	---	---	mesg[log]/R	var	scalar
egvc_pos_govout_w	---	---	mesg[cont]/R	var	scalar
fqsc_s_s	---	---	mesg[cont]/R	var	scalar

Name	Memory	Comment
derp_r_w	v	---
eahs_r_kcgin_w	v	---
eahs_t_ambtemp_w	v	---
eams_dm_prsrlf_w	v	---
eapp_dpv_w	v	---
eapp_pv_w	v	---
eaps_p_w	v	---
ebps_p_w	v	---
ecdt_t_w	v	---
eess_n_avg_w	v	---
egdm_dm_aftsteplim_w	v	---
egdm_dm_des_w	v	---
egfm_dm_atk_w	v	---
egfm_dm_egrout_w	v	---
egoo_s_allowed_b	v	internal egoo calculation because bypass kills output of module calculation
egvc_pos_govout_w	v	---
fqsc_s_s	v	---

2.3.5 Exported Elements

Name	Value	Unit	Modeltype	Kind	Dimension
rmtc_d_rv_diou_ul	0.0	---	mesg[cont]/S	var	scalar
rmtc_dm_egdm_w	0.0	---	mesg[cont]/S	var	scalar

Name	Memory	Comment
rmtc_d_rv_diou_ul	v	bypass insertion of EGOO
rmtc_dm_egdm_w	v	---

2.3.6 Implementation for exported elements

Name	Model(M)				Formula	Implementation(I)		
	Type	Min.	Max.	Q.	M=a*I+b	Type	Min.	Max.
rmtc_d_rv_diou_ul	cont	-oo	+oo	0	ident	real64	-oo	+oo
rmtc_dm_egdm_w	cont	-oo	+oo	0	ident	real64	-oo	+oo

2.3.7 Processes

process []

Init []

2.3.8 Diagrams and Hierarchies

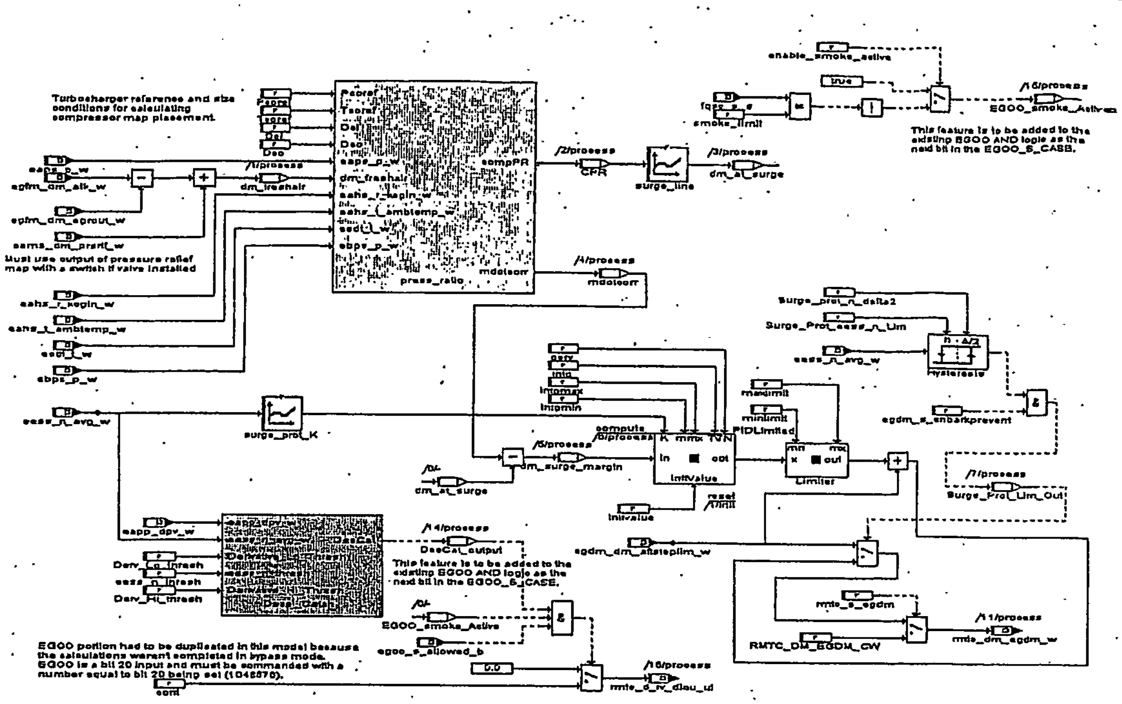

Figure 4: Diagram for the processes

2.3.9 Implementations

Name: Impl

Notes:

Elements:

Name	Model(M)				Formula	Implementation(I)		
	Type	Min.	Max.	Q.	M=a*I+b	Type	Min.	Max.
cont	cont	-oo	+oo	0	ident	real64	-oo	+oo
CPR	cont	-oo	+oo	0	ident	real64	-oo	+oo
Dci	cont	-oo	+oo	0	ident	real64	-oo	+oo
Dco	cont	-oo	+oo	0	ident	real64	-oo	+oo
derv	cont	-oo	+oo	0	ident	real64	-oo	+oo
Derv_Hi_thresh	cont	-oo	+oo	0	ident	real64	-oo	+oo
Derv_Lo_thresh	cont	-oo	+oo	0	ident	real64	-oo	+oo
dm_at_surge	cont	-oo	+oo	0	ident	real64	-oo	+oo
dm_freshair	cont	-oo	+oo	0	ident	real64	-oo	+oo
dm_surge_margin	cont	-oo	+oo	0	ident	real64	-oo	+oo
eess_n_thresh	cont	-oo	+oo	0	ident	real64	-oo	+oo

initvalue	cont	-oo	+oo	0	ident	real64	-oo	+oo
intg	cont	-oo	+oo	0	ident	real64	-oo	+oo
intgmax	cont	-oo	+oo	0	ident	real64	-oo	+oo
intgmin	cont	-oo	+oo	0	ident	real64	-oo	+oo
maxlimit	cont	-oo	+oo	0	ident	real64	-oo	+oo
mdotcorr	cont	-oo	+oo	0	ident	real64	-oo	+oo
minlimit	cont	-oo	+oo	0	ident	real64	-oo	+oo
Pcoref	cont	-oo	+oo	0	ident	real64	-oo	+oo
remote_vpsp	cont	-oo	+oo	0	ident	real64	-oo	+oo
RMTC_DM_EGDM_CW	cont	-oo	+oo	0	ident	real64	-oo	+oo
smoke_limit	cont	-oo	+oo	0	ident	real64	-oo	+oo
surge_line(x)	cont	-oo	+oo	0	ident	real64	-oo	+oo
surge_line(v)	cont	-oo	+oo	0	ident	real64	-oo	+oo
Surge_Prot_eess_n_Lim	cont	-oo	+oo	0	ident	real64	-oo	+oo
surge_prot_K(x)	cont	-oo	+oo	0	ident	real64	-oo	+oo
surge_prot_K(v)	cont	-oo	+oo	0	ident	real64	-oo	+oo
Surge_prot_n_delta2	cont	-oo	+oo	0	ident	real64	-oo	+oo
Tcoref	cont	-oo	+oo	0	ident	reai64	-oo	+oo

Name	Implementation_type	Name	Implementation_type
DecCat_output	int8	egdm_s_enbarkprevent	int8
EGOO_smoke_Active	int8	enable_smoke_active	int8
rmtc_s_egdm	int8	Surge_Prot_Lim_Out	int8

3 ETAS_SystemLib_CT

3.1 MathFcn

Library : Component Library ETAS_SystemLib_CT
Entry : Component MathFcn

Path:

ETAS_SystemLib_CT/Classes/MathFcn

Notes:

3.1.1 Layout

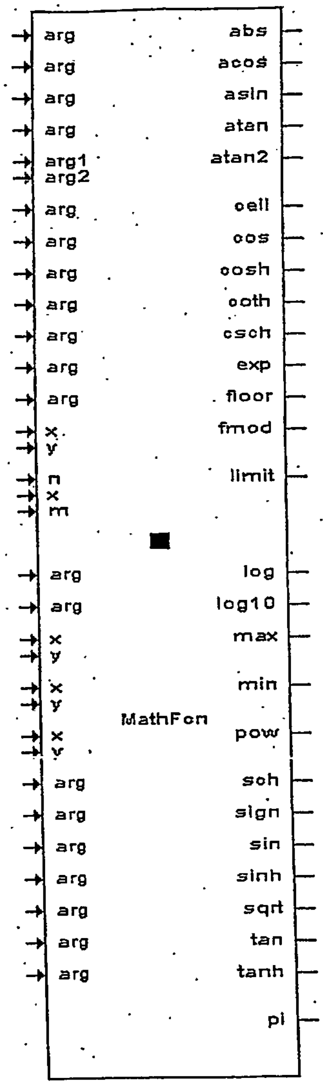

Figure 5: Layout of class MathFcn

3.1.2 Public Methods

exp

Name	Type	Unit	Comment	Kind
exp/arg	cont	---	---	arg
exp/return	cont	---	---	ret

sqrt

Name	Type	Unit	Comment	Kind
sqrt/arg	cont	---	---	arg
sqrt/return	cont	---	---	ret

abs

Name	Type	Unit	Comment	Kind
abs/arg	cont	---	---	arg
abs/return	cont	---	---	ret

atan

Name	Type	Unit	Comment	Kind
atan/arg	cont	---	---	arg
atan/return	cont	---	---	ret

acos

Name	Type	Unit	Comment	Kind
acos/arg	cont	---	---	arg
acos/return	cont	---	---	ret

sin

Name	Type	Unit	Comment	Kind
sin/arg	cont	---	---	arg
sin/return	cont	---	---	ret

cos

Name	Type	Unit	Comment	Kind
cos/arg	cont	---	---	arg
cos/return	cont	---	---	ret

asin

Name	Type	Unit	Comment	Kind
asin/arg	cont	---	---	arg
asin/return	cont	---	---	ret

sinh

Name	Type	Unit	Comment	Kind
sinh/arg	cont	---	---	arg
sinh/return	cont	---	---	ret

cosh

Name	Type	Unit	Comment	Kind
cosh/arg	cont	---	---	arg
cosh/return	cont	---	---	ret

csch

Name	Type	Unit	Comment	Kind
csch/arg	cont	---	---	arg
csch/return	cont	---	---	ret

coth

Name	Type	Unit	Comment	Kind
coth/arg	cont	---	---	arg
coth/return	cont	---	---	ret

sch

Name	Type	Unit	Comment	Kind
sch/arg	cont	---	---	arg
sch/return	cont	---	---	ret

tanh

Name	Type	Unit	Comment	Kind
tanh/arg	cont	---	---	arg
tanh/return	cont	---	---	ret

sign

Name	Type	Unit	Comment	Kind
sign/arg	cont	---	---	arg
sign/return	cont	---	---	ret

pi

Name	Type	Unit	Comment	Kind
pi/return	cont	---	---	ret

tan

Name	Type	Unit	Comment	Kind
tan/arg	cont	---	---	arg
tan/return	cont	---	---	ret

log10

Name	Type	Unit	Comment	Kind
log10/arg	cont	---	---	arg
log10/return	cont	---	---	ret

min

Name	Type	Unit	Comment	Kind
min/x	cont	---	---	arg
min/y	cont	---	---	arg
min/return	cont	---	---	ret

max

Name	Type	Unit	Comment	Kind
max/x	cont	---	---	arg
max/y	cont	---	---	arg
max/return	cont	---	---	ret

pow

Name	Type	Unit	Comment	Kind
pow/x	cont	---	---	arg
pow/y	cont	---	---	arg
pow/return	cont	---	---	ret

limit

Name	Type	Unit	Comment	Kind
limit/n	cont	---	---	arg
limit/x	cont	---	---	arg
limit/m	cont	---	---	arg
limit/return	cont	---	---	ret

fmod

Name	Type	Unit	Comment	Kind
fmod/x	cont	---	---	arg
fmod/y	cont	---	---	arg
fmod/return	cont	---	---	ret

atan2

Name	Type	Unit	Comment	Kind
atan2/arg1	cont	---	---	arg
atan2/arg2	cont	---	---	arg
atan2/return	cont	---	---	ret

ceil

Name	Type	Unit	Comment	Kind
ceil/arg	cont	---	---	arg
ceil/return	cont	---	---	ret

floor

Name	Type	Unit	Comment	Kind
floor/arg	cont	---	---	arg
floor/return	cont	---	---	ret

log

Name	Type	Unit	Comment	Kind
log/arg	cont	---	---	arg
log/return	cont	---	---	ret

3.1.3 Sources

Target: ES1130

no code specified

3.1.4 Implementations

Name: Impl

Notes:

Elements:

Name	Model(M)				Formula	Implementation(I)		
	Type	Min.	Max.	Q.	M=a*I+b	Type	Min.	Max.
abs/arg	cont	-oo	+oo	0	default	real64	-oo	+oo
abs/return	cont	-oo	+oo	0	default	real64	-oo	+oo
acos/arg	cont	-oo	+oo	0	default	real64	-oo	+oo
acos/return	cont	-oo	+oo	0	default	real64	-oo	+oo
asin/arg	cont	-oo	+oo	0	default	real64	-oo	+oo
asin/return	cont	-oo	+oo	0	default	real64	-oo	+oo
atan/arg	cont	-oo	+oo	0	default	real64	-oo	+oo
atan/return	cont	-oo	+oo	0	default	real64	-oo	+oo
atan2/arg1	cont	-oo	+oo	0	default	real64	-oo	+oo
atan2/arg2	cont	-oo	+oo	0	default	real64	-oo	+oo
atan2/return	cont	-oo	+oo	0	default	real64	-oo	+oo
ceil/arg	cont	-oo	+oo	0	default	real64	-oo	+oo
ceil/return	cont	-oo	+oo	0	default	real64	-oo	+oo
cos/arg	cont	-oo	+oo	0	default	real64	-oo	+oo
cos/return	cont	-oo	+oo	0	default	real64	-oo	+oo
cosh/arg	cont	-oo	+oo	0	default	real64	-oo	+oo
cosh/return	cont	-oo	+oo	0	default	real64	-oo	+oo
coth/arg	cont	-oo	+oo	0	default	real64	-oo	+oo
coth/return	cont	-oo	+oo	0	default	real64	-oo	+oo
csch/arg	cont	-oo	+oo	0	default	real64	-oo	+oo
csch/return	cont	-oo	+oo	0	default	real64	-oo	+oo

exp/arg	cont	-oo	+oo	0	default	real64	-oo	+oo
exp/return	cont	-oo	+oo	0	default	real64	-oo	+oo
floor/arg	cont	-oo	+oo	0	default	real64	-oo	+oo
floor/return	cont	-oo	+oo	0	default	real64	-oo	+oo
fmod/return	cont	-oo	+oo	0	default	real64	-oo	+oo
fmod/x	cont	-oo	+oo	0	default	real64	-oo	+oo
fmod/y	cont	-oo	+oo	0	default	real64	-oo	+oo
limit/m	cont	-oo	+oo	0	default	real64	-oo	+oo
limit/n	cont	-oo	+oo	0	default	real64	-oo	+oo
limit/return	cont	-oo	+oo	0	default	real64	-oo	+oo
limit/x	cont	-oo	+oo	0	default	real64	-oo	+oo
log/arg	cont	-oo	+oo	0	default	real64	-oo	+oo
log/return	cont	-oo	+oo	0	default	real64	-oo	+oo
log10/arg	cont	-oo	+oo	0	default	real64	-oo	+oo
log10/return	cont	-oo	+oo	0	default	real64	-oo	+oo
max/return	cont	-oo	+oo	0	default	real64	-oo	+oo
max/x	cont	-oo	+oo	0	default	real64	-oo	+oo
max/y	cont	-oo	+oo	0	default	real64	-oo	+oo
min/return	cont	-oo	+oo	0	default	real64	-oo	+oo
min/x	cont	-oo	+oo	0	default	real64	-oo	+oo
min/y	cont	-oo	+oo	0	default	real64	-oo	+oo
pi/return	cont	-oo	+oo	0	default	real64	-oo	+oo
pow/return	cont	-oo	+oo	0	default	real64	-oo	+oo
pow/x	cont	-oo	+oo	0	default	real64	-oo	+oo
pow/y	cont	-oo	+oo	0	default	real64	-oo	+oo
sch/arg	cont	-oo	+oo	0	default	real64	-oo	+oo
sch/return	cont	-oo	+oo	0	default	real64	-oo	+oo
sign/arg	cont	-oo	+oo	0	default	real64	-oo	+oo
sign/return	cont	-oo	+oo	0	default	real64	-oo	+oo
sin/arg	cont	-oo	+oo	0	default	real64	-oo	+oo
sin/return	cont	-oo	+oo	0	default	real64	-oo	+oo
sinh/arg	cont	-oo	+oo	0	default	real64	-oo	+oo
sinh/return	cont	-oo	+oo	0	default	real64	-oo	+oo
sqrt/arg	cont	-oo	+oo	0	default	real64	-oo	+oo
sqrt/return	cont	-oo	+oo	0	default	real64	-oo	+oo
tan/arg	cont	-oo	+oo	0	default	real64	-oo	+oo
tan/return	cont	-oo	+oo	0	default	real64	-oo	+oo
tanh/arg	cont	-oo	+oo	0	default	real64	-oo	+oo
tanh/return	cont	-oo	+oo	0	default	real64	-oo	+oo

4 ETAS_SystemLib_SD_discrete

4.1 Hysteresis_MSP_DeltaHalf

Library	:	Component Library	ETAS_SystemLib_SD_discrete
Entry	:	Component	Hysteresis_MSP_DeltaHalf

Path:

ETAS_SystemLib_SD_discrete/Nonlinears/Hysteresis_MSP_DeltaHalf

Notes:

default_T

```
Hysteresis-MSP-DeltaHalf is a hysteresis with a middle switching point and a
delta/2 offset.

Methods:
         out:      Arguments:    x::continuous
                                 msp::continuous
                                 deltahalf::continuous
                   Return Value: logical

On activation of method
         out:  TRUE is returned, if x > (msp + deltahalf).
               FALSE is returned, if x < (msp - deltahalf).
               The Return Value is unchanged, if input x lies within the open
               interval ](msp - deltahalf), (msp + deltahalf)[.
```

4.1.1 Layout

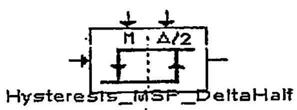

Figure 6: Layout of class Hysteresis_MSP_DeltaHalf

4.1.2 Elements

Name	Value	Unit	Modeltype	Kind	Dimension
hysterese	false	---	log	var	scalar

Name	Memory	Comment
hysterese	v	---

4.1.3 Public Methods

out

Name	Type	Unit	Comment	Kind
out/x	cont	---	---	arg
out/msp	cont	---	---	arg
out/deltaHalf	cont	---	---	arg
out/return	log	---	---	ret

4.1.4 Diagrams and Hierarchies

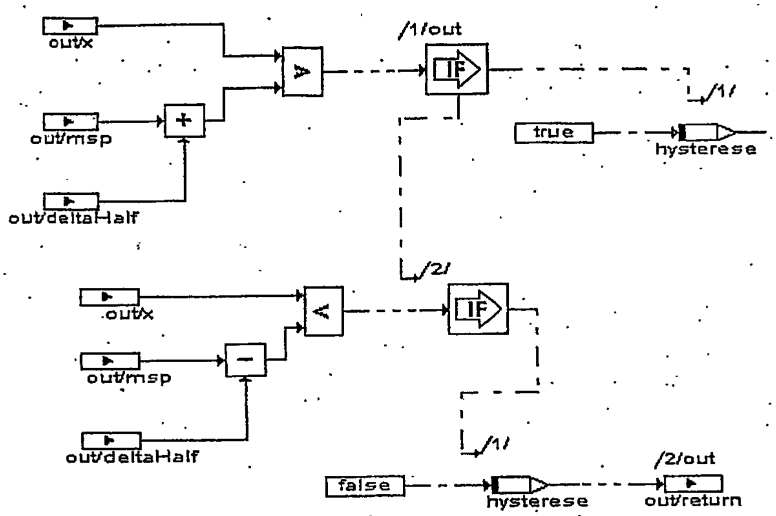

Figure 7: out

4.1.5 Implementations

Name: Impl

Notes:

—

Elements:

Name	Model(M)				Formula	Implementation(I)		
	Type	Min.	Max.	Q.	M=a*I+b	Type	Min.	Max.
out/deltaHalf	cont	-oo	+oo	0	default	real64	-oo	+oo
out/msp	cont	-oo	+oo	0	default	real64	-oo	+oo
out/x	cont	-oo	+oo	0	default	real64	-oo	+oo

Name	Implementation_type	Name	Implementation_type
hysterese	int8	out/return	int8

Name: S16

Notes:

—

Elements:

Name	Model(M)				Formula	Implementation(I)		
	Type	Min.	Max.	Q.	M=a*I+b	Type	Min.	Max.

out/deltaHalf	cont	-32768.0	32767.0	1.0	default	int16	-32768	32767
out/msp	cont	-32768.0	32767.0	1.0	default	int16	-32768	32767
out/x	cont	-32768.0	32767.0	1.0	default	int16	-32768	32767

Name	Implementation_type	Name	Implementation_type
hysterese	uint8	out/return	uint8

Name: S32

Notes:

Elements:

Name	Model(M)				Formula	Implementation(I)		
	Type	Min.	Max.	Q.	M=a*I+b	Type	Min.	Max.
out/deltaHalf	cont	-2147483648.0	2147483647.0	1.0	default	int32	-2147483648	2147483647
out/msp	cont	-2147483648.0	2147483647.0	1.0	default	int32	-2147483648	2147483647
out/x	cont	-2147483648.0	2147483647.0	1.0	default	int32	-2147483648	2147483647

Name	Implementation_type	Name	Implementation_type
hysterese	uint8	out/return	uint8

Name: S8

Notes:

Elements:

Name	Model(M)				Formula	Implementation(I)		
	Type	Min.	Max.	Q.	M=a*I+b	Type	Min.	Max.
out/deltaHalf	cont	-128.0	127.0	1.0	default	int8	-128	127
out/msp	cont	-128.0	127.0	1.0	default	int8	-128	127
out/x	cont	-128.0	127.0	1.0	default	int8	-128	127

Name	Implementation_type	Name	Implementation_type
hysterese	uint8	out/return	uint8

Name: U16

Notes:

Elements:

Name	Model(M)				Formula	Implementation(I)		
	Type	Min.	Max.	Q.	M=a*I+b	Type	Min.	Max.
out/deltaHalf	cont	0.0	65535.0	1.0	default	uint16	0	65535
out/msp	cont	0.0	65535.0	1.0	default	uint16	0	65535
out/x	cont	0.0	65535.0	1.0	default	uint16	0	65535

Name	Implementation_type	Name	Implementation_type
hysterese	uint8	out/return	uint8

Name: U32

Notes:

Elements:

Name	Model(M)				Formula	Implementation(I)		
	Type	Min.	Max.	Q.	M=a*I+b	Type	Min.	Max.
out/deltaHalf	cont	0.0	4294967295.0	1.0	default	uint32	0	4294967295
out/msp	cont	0.0	4294967295.0	1.0	default	uint32	0	4294967295
out/x	cont	0.0	4294967295.0	1.0	default	uint32	0	4294967295

Name	Implementation_type	Name	Implementation_type
hysterese	uint8	out/return	uint8

Name: U8

Notes:

Elements:

Name	Model(M)				Formula	Implementation(I)		
	Type	Min.	Max.	Q.	M=a*I+b	Type	Min.	Max.
out/deltaHalf	cont	0.0	255.0	1.0	Identity	uint8	0	255
out/msp	cont	0.0	255.0	1.0	Identity	uint8	0	255
out/x	cont	0.0	255.0	1.0	Identity	uint8	0	255

Name	Implementation_type	Name	Implementation_type
hysterese	uint8	out/return	uint8

4.2 Limiter

Library	:	Component Library	ETAS_SystemLib_SD_discrete
Entry	:	Component	Limiter

Path:

ETAS_SystemLib_SD_discrete/Nonlinears/Limiter

Notes:

default_T

```
Limiter returns the input x limited by mn and mx.

Methods:
        out:      Arguments:     x::continuous
                                 mn::continuous
                                 mx::continuous
                  Return Value:  continuous

On activation of method
        out:  The input x is limited by mn and mx and is returned, i.e, max(
min(x, mx), mn).
              There is no check if mn <= mx.
```

4.2.1 Layout

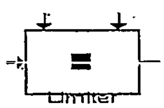

Figure 8: Layout of class Limiter

4.2.2 Public Methods

out

Name	Type	Unit	Comment	Kind
out/mn	cont	—	—	arg
out/x	cont	—	—	arg
out/mx	cont	—	—	arg
out/return	cont	—	—	ret

4.2.3 Diagrams and Hierarchies

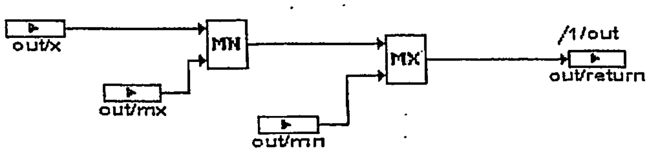

Figure 9: out

4.2.4 Implementations

Name: Impl

Notes:

Elements:

Name	Model(M)				Formula	Implementation(I)		
	Type	Min.	Max.	Q.	M=a*I+b	Type	Min.	Max.
out/mn	cont	-oo	+oo	0	default	real64	-oo	+oo
out/mx	cont	-oo	+oo	0	default	real64	-oo	+oo
out/return	cont	-oo	+oo	0	default	real64	-oo	+oo
out/x	cont	-oo	+oo	0	default	real64	-oo	+oo

Name: S16

Notes:

Elements:

Name	Model(M)				Formula	Implementation(I)		
	Type	Min.	Max.	Q.	M=a*I+b	Type	Min.	Max.
out/mn	cont	-32768.0	32767.0	1.0	default	int16	-32768	32767
out/mx	cont	-32768.0	32767.0	1.0	default	int16	-32768	32767
out/return	cont	-32768.0	32767.0	1.0	default	int16	-32768	32767
out/x	cont	-32768.0	32767.0	1.0	default	int16	-32768	32767

Name: S32

Notes:

Elements:

Name	Model(M)				Formula	Implementation(I)		
	Type	Min.	Max.	Q.	M=a*I+b	Type	Min.	Max.
out/mn	cont	-2147483648.0	2147483647.0	1.0	default	int32	-2147483648	2147483647
out/mx	cont	-2147483648.0	2147483647.0	1.0	default	int32	-2147483648	2147483647
out/return	cont	-2147483648.0	2147483647.0	1.0	default	int32	-2147483648	2147483647
out/x	cont	-2147483648.0	2147483647.0	1.0	default	int32	-2147483648	2147483647

Name: S8

Notes:

Elements:

Name	Model(M)				Formula	Implementation(I)		
	Type	Min.	Max.	Q.	M=a*I+b	Type	Min.	Max.
out/mn	cont	-128.0	127.0	1.0	default	int8	-128	127
out/mx	cont	-128.0	127.0	1.0	default	int8	-128	127
out/return	cont	-128.0	127.0	1.0	default	int8	-128	127
out/x	cont	-128.0	127.0	1.0	default	int8	-128	127

Name: U16

Notes:

Elements:

Name	Model(M)				Formula	Implementation(I)		
	Type	Min.	Max.	Q.	M=a*I+b	Type	Min.	Max.
out/mn	cont	0.0	65535.0	1.0	default	uint16	0	65535
out/mx	cont	0.0	65535.0	1.0	default	uint16	0	65535
out/return	cont	0.0	65535.0	1.0	default	uint16	0	65535
out/x	cont	0.0	65535.0	1.0	default	uint16	0	65535

Name: U32

Notes:

Elements:

Name	Model(M)				Formula	Implementation(I)		
	Type	Min.	Max.	Q.	M=a*I+b	Type	Min.	Max.
out/mn	cont	0.0	42949672 95.0	1.0	default	uint32	0	42949672 95
out/mx	cont	0.0	42949672 95.0	1.0	default	uint32	0	42949672 95
out/return	cont	0.0	42949672 95.0	1.0	default	uint32	0	42949672 95
out/x	cont	0.0	42949672 95.0	1.0	default	uint32	0	42949672 95

Name: U8

Notes:

Elements:

Name	Model(M)				Formula	Implementation(I)		
	Type	Min.	Max.	Q.	M=a*I+b	Type	Min.	Max.
out/mn	cont	0.0	255.0	1.0	Identity	uint8	0	255
out/mx	cont	0.0	255.0	1.0	Identity	uint8	0	255

out/return	cont	0.0	255.0	1.0	Identity	uint8	0	255
out/x	cont	0.0	255.0	1.0	Identity	uint8	0	255

4.3 PIDLimited

Library : Component Library ETAS_SystemLib_SD_discrete
Entry : Component PIDLimited

Path:

ETAS_SystemLib_SD_discrete/Transferfunction/Control/PIDLimited

Notes:

General Comments

PIDLimited is a discrete propotional inegrator with differential part, with time constants Tv
and Tn and a grain constant K. The value of the integrator is limited.

```
Methods:
        reset:    Arguments:     initValue::continuous
                  Return Value:  none
        compute:  Arguments:     in::continuous
                                 Tv::continuous
                                 Tn::continuous
                                 K::continuous
                                 mn::continuous
                                 mx::continuous
                  Return Value:  none
        out:      Arguments:     none
                  Return Value:  continuous

On activation of method
        reset:    The integrator value is set to initValue.
        compute:  The value of the PID-function is computed as a sum of a P-
function
                  a D-function and an I-function, where the integrator value of
the
                  I-function is limited by mn and mx.
        out:      The value of the PID-function is returned.
```

Tracking

Confidence level:

Percent complete:

Open Items:

4.3.1 Layout

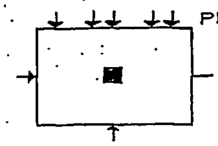

Figure 10: Layout of class PIDLimited

4.3.2 Elements

Name	Value	Unit	Modeltype	Kind	Dimension
inOLD	0.0	---	cont	var	scalar
memory1	0.0	---	cont	var	scalar
memory2	0.0	---	cont	var	scalar

Name	Memory	Comment
inOLD	v	---
memory1	v	---
memory2	v	---

4.3.3 Imported Elements

Name	Value	Unit	Modeltype	Kind	Dimension
dT	---	---	dT	var	scalar

Name	Memory	Comment
dT	v	---

4.3.4 Public Methods

out

Name	Type	Unit	Comment	Kind
out/return	cont	---	---	ret

compute

Name	Type	Unit	Comment	Kind
compute/in	cont	---	---	arg
compute/K	cont	---	---	arg
compute/TV	cont	---	---	arg
compute/TN	cont	---	---	arg
compute/mn	cont	---	---	arg
compute/mx	cont	---	---	arg

reset

Name	Type	Unit	Comment	Kind

reset/initValue	cont	---	---	arg

4.3.5 Diagrams and Hierarchies

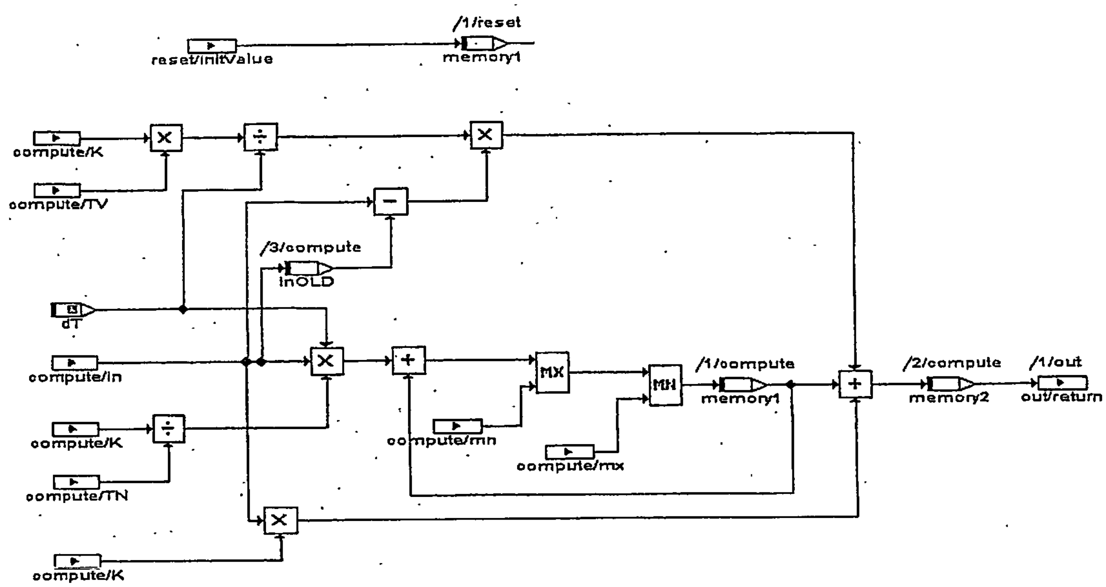

Figure 11: out, compute, reset

4.3.6 Implementations

Name: Impl

Notes:.

Elements:

Name	Model(M)				Formula	Implementation(I)		
	Type	Min.	Max.	Q.	M=a*I+b	Type	Min.	Max.
compute/in	cont	-oo	+oo	0	default	real64	-oo	+oo
compute/K	cont	-oo	+oo	0	default	real64	-oo	+oo
compute/mn	cont	-oo	+oo	0	default	real64	-oo	+oo
compute/mx	cont	-oo	+oo	0	default	real64	-oo	+oo
compute/TN	cont	-oo	+oo	0	default	real64	-oo	+oo
compute/TV	cont	-oo	+oo	0	default	real64	-oo	+oo
inOLD	cont	-oo	+oo	0	default	real64	-oo	+oo

memory1	cont	-oo	+oo	0	default	real64	-oo	+oo
memory2	cont	-oo	+oo	0	default	real64	-oo	+oo
out/return	cont	-oo	+oo	0	default	real64	-oo	+oo
reset/initValue	cont	-oo	+oo	0	default	real64	-oo	+oo

[0027] The foregoing has described the principles, embodiments, and modes of operation of the present invention. However, the invention should not be construed as being limited to the particular embodiments described above, as they should be regarded as being illustrative and not restrictive. It should be appreciated that variations may be made in those embodiments by those skilled in the art without departing from the scope of the present invention.

[0028] While the invention has been described in detail above, the invention is not intended to be limited to the specific embodiments as described. It is evident that those skilled in the art may now make numerous uses and modifications of and departures from the specific embodiments described herein without departing from the inventive concepts.

[0029] While various embodiments of the present invention have been described above, they should be understood to have been presented by way of examples only, and not limitation. Thus, the breadth and scope of the present invention should not be limited by the above described embodiments.

[0030] Obviously, numerous modifications and variations of the present invention are possible in light of the above teachings. It is therefore to be understood that the invention may be practiced otherwise than as specifically described herein.